软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(2):915−933 [doi: 10.13328/j.cnki.jos.007461] [CSTR: 32375.14.jos.007461]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

以用户为中心的云辅助跨应用数据安全流转*

张 源[1], 陈曦露[1], 宋雅晴[1], 高 歌[1], 唐 鹏[2], 邱卫东[2], 李洪伟[1]

[1](电子科技大学 计算机科学与工程学院 (网络空间安全学院), 四川 成都 611731)

[2](上海交通大学 网络空间安全学院, 上海 200240)

通信作者: 宋雅晴, yaqing.song@uestc.edu.cn

摘 要: 提出一种云存储协助下以用户为中心的数据安全流转方案 CADC (user-centric secure cloud-assisted cross-application data circulation scheme), 实现多 App 环境下移动用户的便捷身份认证与海量数据的按需可信流转, 支撑移动互联网环境下的数据价值充分释放. 形式化的安全性分析表明, CADC 能够抵御半诚实的云服务提供商与 App 服务提供商. 仿真实验结果表明, CADC 在用户端与 App 端具有较高的效率.

关键词: 数据安全流转; 身份认证; 云存储; 数据价值释放; 跨应用

中图法分类号: TP309

中文引用格式: 张源, 陈曦露, 宋雅晴, 高歌, 唐鹏, 邱卫东, 李洪伟. 以用户为中心的云辅助跨应用数据安全流转. 软件学报, 2026, 37(2): 915–933. http://www.jos.org.cn/1000-9825/7461.htm

英文引用格式: Zhang Y, Chen XL, Song YQ, Gao G, Tang P, Qiu WD, Li HW. User-centric Secure Cloud-assisted Cross-application Data Circulation. Ruan Jian Xue Bao/Journal of Software, 2026, 37(2): 915–933 (in Chinese). http://www.jos.org.cn/1000-9825/7461.htm

User-centric Secure Cloud-assisted Cross-application Data Circulation

ZHANG Yuan[1], CHEN Xi-Lu[1], SONG Ya-Qing[1], GAO Ge[1], TANG Peng[2], QIU Wei-Dong[2], LI Hong-Wei[1]

[1](School of Computer Science and Engineering (School of Cyber Security), University of Electronic Science and Technology of China, Chengdu 611731, China)

[2](School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: This study proposes a cloud-assisted, user-centric, and secure data circulation scheme, referred to as CADC (user-centric secure cloud-assisted cross-application data circulation scheme). It enables convenient authentication and on-demand trusted data circulation for mobile users in a multi-App environment, allowing the unlocking of data value in mobile networks. Formal security analysis demonstrates that CADC can resist semi-honest cloud service providers and App service providers. Performance evaluation results indicate that CADC is highly efficient for both users and Apps.

Key words: secure data circulation; identity authentication; cloud storage; data value unleashing; cross-application

数据流转指数据在不同系统、平台或环境之间传递和转移的过程, 实现了跨系统、平台或组织的信息传递, 已成为数据技术 (data technology, DT) 时代数据要素价值释放的最重要途径[1,2]. 在移动互联网中, 用户跨应用程序 App (application) 的数据流转与融合利用能够显著提升用户服务体验、催生新兴服务范式, 已经成为促进数字经济发展的重要力量[3,4].

实现以用户为中心的数据跨 App 流转最通用的方式是利用云存储服务[5]. 具体来说, 多个 App 服务提供商使用同一个云服务器存储和管理用户数据, 云服务器为用户和 App 提供数据访问和检索接口[6]. App 可以向用户发

* 基金项目: 国家重点研发计划 (2023YFB3106503); 国家自然科学基金 (62472074); 第八届中国科协青年人才托举工程 (2022QNRC001); 四川省重大科技专项 (2022ZDZX0038); 四川省自然科学基金 (24NSFSC2271)
收稿时间: 2024-04-30; 修改时间: 2024-10-10; 采用时间: 2025-05-03; jos 在线出版时间: 2025-11-26
CNKI 网络首发时间: 2025-11-27

送数据访问请求, 用户在云存储服务器上按需检索并下载其他 App 为其产生的数据, 随后转发给所需 App, 从而实现以用户为中心的数据跨 App 流转[7]. 例如, 用户同时使用视频拍摄软件、视频编辑软件和社交分享平台 App, 假设上述 3 个软件使用同一个云存储服务平台管理用户数据. 用户通过云存储服务可以在视频编辑软件中直接获取视频拍摄软件中的视频素材, 并在社交分享平台上选择编辑完成的视频进行分享. 当前, 基于云存储服务的以用户为中心的数据跨域流转已经在协同办公、影音娱乐等领域得到了广泛的应用, 并取得了较好的经济和社会效益[8−11].

然而, 在云辅助下实现跨 App 的数据流转面临着严峻的安全威胁. 首先, 跨 App 的冒充攻击频发. 敌手可能会冒充合法用户登录 App, 越权访问用户已购买的相关服务, 侵害用户权益; 敌手也可能会冒充用户执行恶意操作, 如发布不良信息等, 进而破坏 App 生态[12]. 其次, 数据泄露问题突出. 用户的 App 使用数据会泄露其兴趣偏好、社会关系和行为习惯等高度敏感的个人隐私数据[13]. 若 App 直接将用户数据文件以明文形式外包存储至云服务器, 用户和 App 会失去对数据的掌控[14], 而云服务器可以对存储数据执行任意操作, 用户数据面临着恶意窃取等威胁. 进一步地, 恶意的 App 可以与云服务器合谋, 从而在未经用户授权的情况下访问用户在其他 App 处的使用数据, 增加用户数据内容泄露的风险. 近期, 一项针对美国和中国应用商店的 300 款下载量最高的 App 的研究分析[15]发现, 超过 55% 的中国应用和 10% 的美国应用在跨 App 数据流转过程中存在泄露用户隐私信息的问题.

认证是保证跨 App 数据流转系统安全的第 1 道防线. 口令认证仍是移动系统中应用最普遍的身份认证方式[16], 然而, 由于口令固有的低熵、与个人信息强相关等特点, 现有口令认证系统易遭受离线字典猜测攻击 (dictionary guessing attack, DGA), 从而导致用户口令泄漏、认证系统失效等严峻的后果[17,18]. 例如, 2019 年, 中国智能家居公司欧瑞博的数据泄漏, 涉及包括用户名、邮箱、口令等内容的 20 亿条日志; 2020 年, CAM4 网站泄露了 108.8 亿个用户口令; 2022 年, NVIDIA 网站泄露 70 万个用户口令 (https://www.forbes.com/sites/kateoflahertyuk/2019/01/17/collection-1-breach-how-to-find-out-if-your-password-has-been-stolen/). 此外, 由于用户倾向于在不同 App 中使用相同或相似的口令, 一旦敌手获取了用户在某一个 App 中的口令, 则其有很大概率获得用户在其他 App 中使用的口令, 进而破坏其他系统的认证安全性. 在移动互联网环境下, 口令认证协议的缺点被进一步放大: 由于移动用户的设备受限、数据访问随遇、访问服务多样等特点, 口令认证协议在实际部署时会极大增加用户的记忆负担和口令维护成本[19]. 因此, 亟需构建适用于移动用户且便携高效的安全认证机制, 支撑数据跨 App 可信流转.

加密算法是防止数据泄露的密码学原语. 然而, 使用传统加密算法会导致用户无法高效按需检索数据. 为了实现在密文上的按需检索, 可采用公钥可搜索加密 (public key encryption with keyword search, PEKS) 机制[20]加密数据. 具体来说, App 使用用户的公钥加密数据的关键词、混合加密算法加密数据内容, 并外包存储至云服务器; 后续数据访问阶段, 用户可以利用私钥对关键词生成陷门, 云服务器以关键词陷门为输入, 执行检索算法并返回包含该陷门对应关键词的所有密文; 最后, 用户解密得到所需数据内容[21]. 云服务器除检索结果外, 无法获得关于查询关键词的任何额外信息. 可搜索加密机制能够同时保证数据的机密性和可用性. 然而, 由于数据文件中关键词的明文空间较小, 关键词普遍是低熵的, 容易遭受离线关键词猜测攻击 (keyword guessing attack, KGA)[22]. 因此, 亟需构建面向移动用户, 抵御离线 KGA 攻击的可搜索加密机制, 保障云辅助下数据跨 App 流转的安全.

我们发现, 用户口令和关键词都具有低熵的性质, 可以使用同一组服务器协助用户对口令和关键词进行强化, 保护口令和关键词的安全性. 在云辅助下数据跨 App 流转的场景中, App 既可以作为身份服务器帮助用户加固长效口令, 又能够作为密钥服务器协助用户强化关键词. 本文首先设计了安全高效的移动用户身份认证方法, 利用多 App 协助的门限口令加固机制和长短效秘密结合的便捷的身份认证机制, 实现了适用于移动用户且便携高效的安全认证机制. 其次, 提出了安全的跨信任域数据流转方法, 利用多 App 协助的公钥可搜索加密机制, 实现了面向移动用户的抵御离线 KGA 攻击的数据安全检索机制. 最后, 将便携高效的安全认证机制与面向移动用户的抵御离线 KGA 攻击的数据安全检索机制进行整合, 形成了以用户为中心的云辅助跨 App 数据安全流转方案 CADC (user-centric secure cloud-assisted cross-application data circulation scheme). 本文提供了 CADC 的形式化安全定义并进行了严格的安全性证明. 仿真实验结果表明, CADC 在用户端与 App 端具有较高的效率.

本文主要的研究工作如下.

(1) 安全高效的移动用户身份认证方法. 使用多 App 协助的门限口令加固机制以及长短效秘密结合的便捷的身份认证机制, 实现适用于移动用户且便携高效的安全认证机制. 用户拥有长效的用户口令和短效的设备公私钥对, 多个 App 作为分布式的身份服务器, 帮助用户加固不可复用的长效用户口令 (长效秘密), 实现在无需可信第三方环境下的身份认证, 避免单点失效问题和离线 DGA 攻击. 完成注册后, 用户使用短效公私钥对 (短效秘密) 授权设备, 通过长短效秘密的结合使用, 在短时间内, 用户只需使用短效私钥即可通过已授权设备登录 App 和云服务器, 无需再次执行口令加固过程.

(2) 安全的跨信任域数据流转方法. 使用多 App 协助的公钥可搜索加密机制实现面向移动用户的抵御离线 KGA 攻击的数据安全检索机制. 各 App 作为分布式的密钥服务器, 分别与用户交互, 帮助用户强化数据检索关键词, 作为服务器的各 App 间无需进行交互, 避免离线 KGA 攻击以及密钥服务器的单点失效问题.由云存储服务器建立加密后的数据文件与对应强化关键词密文的索引表, 通过检验强化关键词密文与陷门的匹配情况检索数据并将数据检索结果返回给用户, 用户将其中允许共享的数据加密后发送给请求数据的 App.

(3) 结合适用于移动用户且便携高效的安全认证机制与面向移动用户的抵御离线 KGA 攻击的数据安全检索机制, 实现了以用户为中心的云辅助跨 App 数据安全流转方案 CADC. 形式化的安全性分析表明, CADC 能够抵御半诚实的云服务提供商与 App 服务提供商. 在仿真实验中, 当用户数达到 500 时, CADC 在 App 端的存储开销不超过 80 KB, 计算开销低于 3.5 s; 在用户端各阶段的通信开销都不超过 10 KB. 仿真实验结果表明, CADC 在用户端与 App 端具有较高的效率.

本文第 1 节介绍相关工作. 第 2 节描述 CADC 需解决的问题, 说明 CADC 的系统模型、定义、安全模型和设计目标. 第 3 节介绍本文提出的 CADC 具体方案, 给出了方案的正确性证明. 第 4 节给出 CADC 的形式化的安全性分析. 第 5 节通过仿真实验分析 CADC 的性能. 最后总结全文.

1 相关工作

1.1 面向移动互联网的安全身份认证技术

在移动互联网环境中, 身份认证的主要任务是确保通信双方或多方能够在不可信网络中确认彼此身份, 并在此基础上建立安全的通信通道[23]. 身份认证过程往往涉及多个实体在开放网络中的交互, 其核心在于通过一系列信息交换与运算步骤, 对通信对象的真实性、合法性和有效性进行判断, 通常依赖用户的个体信息、随身设备以及其固有生物特征来实现身份识别. 基于所使用的认证因子不同, 当前应用于移动互联网的身份认证方法一般可分为 3 类[24]: (1) 基于用户掌握的秘密信息, 例如静态口令和基于知识的认证 (knowledge-based authentication, KBA); (2) 基于用户所持硬件设备, 如硬件令牌、智能卡、手机短信验证以及近场通信技术 (near field communication, NFC); (3) 基于用户独特生物特征的认证方式, 包括指纹、声纹、虹膜以及人脸识别等.

从安全性与可用性角度考量, 这 3 类方法各具有优势与不足[25]. 首先, 基于秘密信息的认证由于部署便捷、成本较低且使用灵活, 已成为最广泛采用的身份认证方法. 在移动应用环境下, 用户只需输入正确信息即可通过服务器验证. 然而, 此类方法的安全性依赖于用户秘密凭据的安全性, 一旦凭据被窃取, 敌手即可轻易冒充合法身份. 其次, 基于硬件设备的认证通常具有更高的安全强度, 因设备能够存储用户难以记忆的敏感密钥. 然而, 此类硬件方案会增加用户负担, 同时设备存在丢失与被盗风险, 一旦设备落入攻击者手中, 系统安全性将面临严重挑战. 最后, 基于生物特征的认证通过自动采集用户的生理或行为特征完成身份确认, 用户无需记忆额外信息或携带硬件设备, 在安全性与便携性之间达成较好平衡[26]. 但由于生物特征不可撤销, 一旦相应特性被非法获取, 将难以通过更新机制进行补救, 从而带来长期安全隐患.

由于口令的便捷性, 基于口令的身份认证是移动互联网中最主要的身份认证方式. 硬件令牌 (或硬件设备) 和生物特征通常作为安全增强因子, 与静态口令结合形成基于口令的多因子身份认证, 以提高身份认证的安全性. 在普通的移动互联网环境 (如绝大多基于网站的服务) 中, 目前主要采用基于口令的单因子身份认证方式; 在一些高安全需求的移动互联网场景 (如电子医疗、电子政务、电子商务和智能家居) 中, 往往采用多因子身份认证技术来实现用户身份认证, 提高系统的安全性[27]. 1981 年, Lamport[28]提出首个适用于客户端-服务器架构的基于口令

的身份认证协议. 但该协议要求用户单独在每个服务器完成注册并记忆其在每个服务器的登录口令, 用户的口令维护成本较高, 在多服务器环境下可用性较低. 1991 年, Chang 等人[29]首次提出了基于口令和智能卡的双因子用户身份认证协议. 但该协议的安全性基于智能卡抗窜扰的强假设, 即假设敌手无法获取智能卡中存储的数据内容. 2001 年, Tsaur[30]提出了首个基于“口令+智能卡”的适于多服务器环境的认证协议, 用户只需要记忆一个口令, 便可登录同一管理域下的多个服务.

随着网络服务规模的不断扩展, 用户需要维护的口令数量急剧攀升, 因记忆负担过重, 用户往往倾向于设置结构简单、易于回忆的弱口令, 并在多个服务间反复使用相同口令[17,18]. 在此情形下, 传统的基于口令的身份认证协议难以抵御离线 DGA 攻击. 为提升口令类凭证的安全性, 自 Lamport 的早期工作提出以来, 国内外研究者陆续构建了大量具备不同防护特性的基于口令的认证方案[25–27]. 最直接的方式是避免在身份服务器中存储口令明文, 而改以存储口令的哈希值. 然而, 当攻击者能够执行大规模离线口令猜测时, 只需重复调用哈希函数即可恢复口令. 因此, 通过采用计算成本更高的新型哈希函数来增加敌手的破解开销, 成为提升认证方案安全性的一个方向[31,32]. 另一类常见方法是引入口令加盐机制[33]: 身份服务器为每个用户维护一个独立盐值, 并存储加盐后的哈希值. 只要盐值不泄露, 敌手便难以利用预计算攻击或离线 DGA 恢复口令. 然而, 盐值完全由身份服务器生成和管理, 一旦半诚实或恶意的服务器泄露盐值, 敌手仍可执行离线 DGA. 相较之下, 服务器协助的基于口令的认证机制[34]在安全性与效率之间取得更好的平衡, 并展现出更高的实际应用价值. 在此类方案中, 用户向一个可信的密钥服务器请求协助, 密钥服务器保存服务器端密钥并与用户交互, 将用户口令“强化”为高强度凭证. 当用户需要向身份服务器证明其身份时, 只需再次通过强化流程生成认证凭证, 由身份服务器比对新旧凭证是否一致即可完成身份认证. 基于上述构建理念, Everspaugh 等人[35]提出了 Pythia 这一基于伪随机函数 (pseudorandom function, PRF) 的代表性方案. 随后, Lai 等人[36]提出了在更强安全性定义下仍保持高效性能的 Phoenix 协议. 然而, Phoenix 依赖身份服务器在整个系统生命周期内始终保持诚实的强假设, 因而容易产生单点失效问题. 为缓解此风险, Agrawal 等人[37]提出了门限化的 PASTA 协议: 借助口令与服务器端秘密共享, 用户在多个服务器协助下生成强化口令与认证令牌, 只要被攻破的服务器数量未超过门限, 方案即可抵御离线 DGA. 但 PASTA 未考虑敌手可自适应选择攻击目标的情形, 也缺乏对服务器故障的修复机制. 基于 PASTA 的框架, Baum 等人[24]进一步设计了分布式单点登录 (single sign on, SSO) 协议 PESTO, 利用不经意的分布式 PRF 与分布式签名技术实现非交互式密钥更新, 并支持敌手自适应攻破服务器. 在被攻破服务器低于门限值的前提下, PESTO 能够在 UC 模型中证明安全性. 然而, 该协议要求用户每次认证时都需经由服务器协助强化口令并生成凭证, 导致通信与计算开销较高, 不利于移动用户的便捷使用. 本文提出的方案 CADC 引入长短效秘密结合的方法, 移动用户拥有长效的用户口令和短效的设备公私钥对, 用户在短期内可直接使用设备私钥完成身份认证, 无需再次与 App 交互以加固长效口令, 极大地提升了用户端的效率, 实现了适用于移动用户且便携高效的安全认证机制.

1.2 数据可信共享与流转

在数据跨 App 流转过程中, 为了抵御半诚实的云服务器, 一种最简单的解决方法是由数据拥有者自行下载全部密文并解密后, 再使用数据接收者的公钥加密数据并发送给数据接收者, 但这无疑给数据拥有者带来了极大的计算开销, 同时也失去了云辅助下完成数据流转的便捷性和高效性. 如何实现云辅助的数据安全跨域流转引起了学术界的广泛关注. 目前, 已有多种技术可以实现不同应用场景下安全的数据跨域流转, 包括隐私集合交集 (private set intersection, PSI) 技术、私有信息检索 (private information retrieval, PIR) 技术、可搜索加密 (searchable encryption) 技术等. 在本节中, 我们讨论了上述技术的应用场景, 以及其在云辅助的跨 App 数据流转场景下应用的优缺点.

PSI 技术能够在双方分别持有私有集合的情况下安全计算其交集, 同时确保除交集外的其他元素不被泄漏[38,39]. 然而, PSI 本质上针对两方计算设计, 对于“以用户为中心”的跨 App 数据流转而言, 用户通常需跨越不同应用查询其数据而非与每个 App 比对上传集合. 如果采用 PSI 完成检索, 用户必须与每一个应用单独运行一次 PSI 协议, 通信与计算代价将指数式增长, 难以满足实际需求. 另一类典型工具是 PIR 技术, 其目标是允许用户在不暴露查询内容的前提下从服务器数据库中检索所需信息[40]. PIR 强调保护查询隐私, 而服务器完全掌握数据库

结构和内容. 在跨 App 数据统一托管于云服务器的条件下, 若单纯采用 PIR 进行数据检索, 可能导致云服务器能够访问所有用户数据, 从而破坏用户侧的隐私保障. 相比之下, 可搜索加密技术为以用户为中心的数据管理提供了更加契合的安全框架[20]. 其基本思想是在用户本地对数据进行加密后外包存储至云服务器, 随后授权用户凭借密钥为查询关键词生成陷门; 陷门不会泄露关键词内容, 而云服务器仅依据陷门执行检索并返回包含该关键词的密文文件. 服务器除了知道某个密文文件是否匹配关键词外, 无法获得额外信息. 用户最终通过解密密文恢复查询结果[21]. 这种模式能够同时保证数据的可用性与保密性, 使用户能够在不暴露数据内容和检索意图的前提下实施高效查询, 因此在跨 App 数据流转场景中比 PSI 与 PIR 方案更具优势. 本研究亦采用可搜索加密作为核心技术路径.

可搜索加密的发展始于 Song 等人[20]的 SWP 方案, 其基于逐词扫描的方式在不可信服务器上实现安全搜索. 尽管 SWP 可确保服务器无法获取多余信息, 但其线性扫描过程使得时间开销随文件规模增大而显著增加, 并因泄露关键词位置而无法抵御统计分析攻击. 随后, Goh[41]在 2003 年提出基于布隆过滤器的 Z-IDX 方案, 而 Chang 等人[42]在 2005 年构建了可抵御统计攻击的 PPSED 方案, 但两者均未显著改善效率问题. Curtmola 等人[43]在 2006 年首次系统化提出对称可搜索加密 (symmetric searchable encryption, SSE) 的安全定义, 并给出适用于非自适应与自适应攻击模型的 SSE-1 与 SSE-2 方案. 围绕提升 SSE 在复杂场景下的性能与安全性问题, 后续研究者陆续提出多种改进的 SSE 方案[44−46].

在不可信服务器的路由环境下, Boneh 等人[47] 率先提出了非对称可搜索加密的概念, 并基于 BF-IBE 构建了首个 PEKS 方案 BDOP-PEKS[48]. 在该机制中, 数据拥有者利用接收者的公钥加密数据及关键词, 接收者凭私钥生成搜索令牌, 服务器据此判断密文是否与令牌匹配. PEKS 常应用于云邮件等存储-转发系统, 其效率低于 SSE 但具有更强的可扩展性.

然而, Byun 等人[22]的研究表明, 真实关键词的明文空间较小且用户习惯使用有限词汇, 导致 PEKS 易受离线 KGA 攻击. 为抵御 KGA, 研究者提出了多类具有不同安全特性的 PEKS 方案[49−54]. 现有方案可以分为 4 类: (1) 依赖可信检测方完成密文与令牌匹配[49], 能够阻止外部攻击者执行 KGA, 但无法抵御半诚实服务器; (2) 引入新型密码学原语, 如不可区分混淆器, 使服务器无法伪造密文进行测试[50], 但混淆器的效率问题限制其实用性; (3) 采用模糊关键词检索, 使服务器难以精确获得关键词[51], 但安全性仍有限且增加用户端负担; (4) 使用服务器协助的 PEKS 方案[53], 引入可信密钥服务器帮助用户生成密文与检索令牌. 与前述方案相比, 服务器协助的 PEKS 方案在安全性与效率间取得更符合实际需求的平衡. 其构建思路源于 Bellare 等人[54]提出的 Dupless 框架, 尽管 Dupless 未被直接应用于 PEKS. 2016 年, Chen 等人[53]构建了首个单服务器协助的 PEKS 方案, 但其假设密钥服务器长期可靠, 因而存在单点失效问题. 2019 年, Zhang 等人[6]提出多服务器协助的 PEKS 机制, 引入分布式密钥服务器及更新机制, 有效缓解 KGA 与单点失效风险, 但未关注适用于移动环境的便携高效的用户身份认证需求. 本文提出的方案 CADC 使用多 App 协助的门限口令加固机制以及长短效秘密结合的便捷的身份认证机制, 在保证云辅助下数据跨 App 流转过程中能够有效抵御离线 KGA 攻击和单点失效问题的基础上, 进一步实现了适用于移动用户且便携高效的安全认证机制, 支撑数据跨 App 可信流转.

2 问题描述

2.1 系统模型

CADC 共包含 3 个实体, 分别是用户、一组 App 和云存储服务器, CADC 的系统模型如图 1 所示.

1) 用户: 用户拥有一个设备、一个不可复用的长效口令和用户名, 用户有一组需要使用的 App. 在身份认证阶段, 用户需要在 App 的协助下加固长效口令, 完成注册. 随后, 用户使用加固后的长效口令申请一个认证证书授权设备, 通过已授权设备, 用户可以在 App 和云存储服务器进行高效的登录和使用. 在数据请求阶段, 用户分别与作为密钥服务器的 App 进行交互, 在各 App 的协助下强化关键词, 计算陷门以完成数据检索, 解密相应数据文件的密文后, 用户可以将其中允许共享的数据发送给请求数据的 App.

2) App: CADC 涉及一组 App $\{A_1, A_2, \ldots, A_n\}$. 它们既作为身份服务器也作为密钥服务器, 一方面, App 分别与用户交互, 协助用户加固长效口令, 通过检查用户是否拥有正确的长效口令来验证其身份. 另一方面, App 分别与

用户或发送数据的 App 交互, 协助用户或发送数据的 App 生成用于数据检索的强化关键词. 作为用户数据的发送者, App 发送数据文件密文与对应强化关键词密文至同一个云存储服务器; 作为数据的请求者, App 向用户发送关键词以请求数据检索.

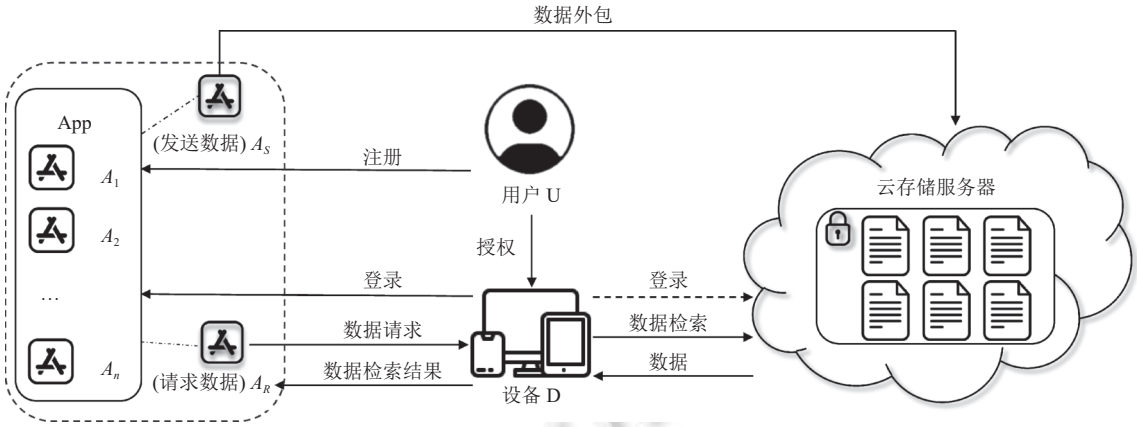


图 1 CADC 系统模型

3) 云存储服务器: 云存储服务器为 App 提供数据存储服务. 云存储服务器接收 App 发送的数据文件密文与对应强化关键词的密文, 建立索引表. 在数据请求阶段, 云存储服务器为用户提供安全高效的通过关键词陷门检索相应数据文件密文的方式, 并将相应目标数据文件的密文发送给用户设备.

CADC 由系统初始化、注册、设备授权、登录、数据外包存储和数据请求这 6 个阶段组成.

1) 系统初始化阶段: 根据安全参数进行系统初始化, 确定系统的公共参数.

2) 注册阶段: App 执行分布式秘密共享协议获取服务器端密钥共享, 用户分别与各 App 交互, 在 App 的协助下加固长效口令, 分别计算各 App 的凭证并发送给对应 App, App 存储该用户的用户名和凭证完成用户注册.

3) 设备授权阶段: 各 App 执行分布式秘密共享协议获取各自的秘密共享份额及公共秘密, 用户生成安全的短效公私钥对并存储在待授权设备上, 在 App 的协助下计算设备证书完成设备授权.

4) 登录阶段: 设备凭设备证书向 App 获取账户证书, 生成与某个 App 的会话证书和安全的短效会话公私钥对, 完成在该 App 的登录.

5) 数据外包存储阶段: 发送数据的 App 分别与各 App 交互, 在其余 App 的帮助下生成强化关键词, 将用户的使用数据和强化关键词加密后发送至云存储服务器, 云存储服务器建立数据文件密文与对应强化关键词密文的索引表.

6) 数据请求阶段: App 向授权设备发送包含关键词的数据请求, 设备分别与各 App 交互, 在 App 的帮助下生成强化关键词, 计算陷门后发送给云存储服务器, 云存储服务器对索引表进行检索并返回对应数据文件的密文, 由用户解密后将其中允许共享的数据发送给请求数据的 App.

2.2 CADC 方案定义

CADC 共包含以下 9 个多项式时间算法.

1) $Setup(\ell)\rightarrow pp$. 系统初始化算法, 根据系统安全参数 ℓ, 确定系统的公共参数集 pp.

2) $MasterKeyGen(\ell,n,t)\rightarrow(s,PS,s_i,PS_i)$. 主密钥生成算法, 由作为分布式服务器的 App 运行. 算法输入系统安全参数 ℓ, 服务器总数 n 和门限值 t, 输出主公私钥对和各 App 的子公私钥对.

3) $Sign(x,r)\rightarrow\sigma$. 签名算法, 由 App 运行. 算法输入待签名数据 x 和随机数 r, 输出签名 σ.

4) $Encap(\sigma_1,\sigma_2,\ldots,\sigma_t)\rightarrow\sigma$. 签名封装算法, 由用户运行. 算法输入门限 t 个有效的签名, 计算并输出封装签名 σ.

5) $RetriSerPsw(\sigma_u,psw_u)\rightarrow sp_u$. App 派生密钥生成算法, 由用户运行. 算法输入封装的盲化口令签名 σ_u 和用

户口令 psw_u, 输出 App 派生密钥 sp_u.

6) $Auth(sp_i, K_D, K_R) \rightarrow Aut_i$. 设备授权算法, 由作为分布式的身份服务器的 App 运行. 算法输入用户凭证 sp_i 和设备公钥 K_D, K_R, 输出设备证书签名 Aut_i.

7) $RetriSerKw(\sigma_w, w) \rightarrow sd_w$. 强化关键词算法, 由发送数据的 App 运行. 算法输入封装的盲化关键词签名 σ_w 和关键词 w, 输出强化关键词 sd_w.

8) $Trapdoor(w) \rightarrow td_w$. 陷门计算算法, 由已授权设备和作为分布式的密钥服务器的 App 运行. 算法输入关键词 w, 输出用于数据检索的陷门 td_w.

9) $Test(td_w) \rightarrow C_{\text{data}}$. 数据检索算法, 由云存储服务器运行. 算法输入陷门 td_w, 输出与其匹配的数据密文 C_{data}.

CADC 的交互流程图如图 2 所示.

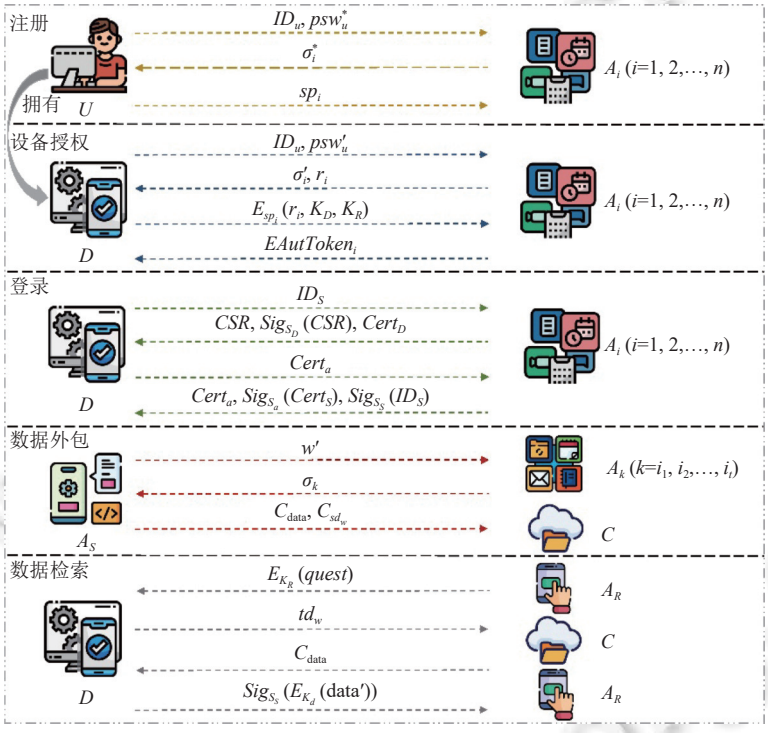


图 2 CADC 交互流程图

2.3 安全模型

基于第 2.1 节的系统模型, 我们考虑以下威胁模型: 用户是一个诚实的实体, 而 App 和云存储服务器是半诚实的实体. 具体地, 在 CADC 中, App 和云存储服务器将诚实地为用户提供相关服务, 但云存储服务器可能会窃听用户与 App 的交互信息, App 可能会与云存储服务器或其他 App 合谋并泄露其秘密份额.

CADC 的安全目标是保证用户长效口令和数据检索关键词的机密性. CADC 的安全性定义如下.

定义 1 (**可靠性**). 当且仅当不存在一个概率多项式时间 (probabilistic polynomial time, PPT) 算法, 敌手 A 能够以不可忽略的优势 (即 $\Pr[CADCsound_{\text{A}}(\ell) = 1]$) 赢得 CADCsound 安全游戏时, CADC 满足可靠性, 其中 ℓ 是安全参数, CADCsound 安全游戏定义如下.

CADCsound: 环境 ε 初始化 CADC 并将 A 作为子程序运行. A 可以访问以下预言机.

1) $MasterKeyGen(\ell, n, t) \rightarrow (s, PS, s_i, PS_i)$: A 适应性地选择 t' ($t' < t$) 个索引 $T = \{i_1, i_2, \ldots, i_{t'}\}$ 进行查询, 预言机输出 App 端公钥和对应 t' 个 App 的秘密份额 $\{k_{i_1}, k_{i_2}, \ldots, k_{i_{t'}}\}$.

2) $OSign(x,r)\rightarrow\sigma$: 对于 $i\in[1,n]/T$, A 选择 x,r 进行查询, 预言机输出签名 σ_i, A 的请求次数为 q.

3) $ORetriSerPsw(c)\rightarrow sp_c$: A 适应性地选择 c 进行查询, 预言机输出 App 派生密钥 sp_c.

4) $ORetriSerKw(w)\rightarrow C_{sd_w}$: A 适应性地选择关键词 w, 预言机输出对应的强化关键词密文 C_{sd_w}.

5) $OTrapdoor(w)\rightarrow td_w$: A 适应性地选择关键词 w, 预言机输出对应的陷门 td_w.

A 随机选取 c_0,c_1,w_0,w_1 发送给 ε. ε 随机选取 $b\in\{0,1\}$ 并输出 $C_{sd_{w_b}},td_{w_b},\sigma_1$ (A_1 对 w_b 的签名); 若 $b=0$, 输出 sp_{c_b}; 若 $b=1$, 输出 $sp_{c_b}=F'(c_b),F'\in Func_{Z_p}$. 若 A 能够输出 c 的有效签名且 $q<t-t'$, 或 A 对 b 的猜测 b' 满足 $|\Pr[b'-b]-1/2|$ 不可忽略, 则 A 在 CADCsound 中获胜, $CADCsound_{\mathrm{A}}(\ell)=1$; 否则, A 在 CADCsound 中失败, $CADCsound_{\mathrm{A}}(\ell)=0$. A 在 CADCsound 中获胜的概率定义为 $\Pr[CADCsound_{\mathrm{A}}(\ell)=1]$.

2.4 设计目标

本文的目标是构建一种云辅助的面向跨信任域 App 的数据流转的安全保护方案, 需要设计 CADC 系统实现以下目标.

1) 功能性. App 能够对用户进行身份验证, 用户只需要维护长效口令即可授权设备并使用设备通过 App 的认证, 使用相关服务; 在用户的授权下, 用户数据能够在各 App 间流转, App 可向用户请求数据流转, 由用户检索数据并将其中允许共享的数据发送给请求数据的 App; 云存储服务器可以通过陷门与检索关键词密文的匹配检索数据.

2) 安全性. CADC 应当能够有效抵御来自敌手的窃听攻击、离线 DGA 攻击和离线 KGA 攻击. 破坏一个或多个 (不超过门限值) App 无法破坏 CADC 的安全.

3) 高效性. 用户的计算开销和通信开销应尽可能小, 各 App 和云存储服务器的存储开销应尽可能低.

3 具体方案

3.1 方案构建

$Setup(\ell)\rightarrow pp$: 系统初始化算法. 在系统初始化阶段, 根据安全参数 ℓ, 确定系统的公共参数集 $PP=\{p,P,G,G_T,e,h,h_1,h_2,H,H_1,H_2,E,D,Sig,F_1,F_2,t,n\}$. 其中, G 是一个加法循环群, G_T 是一个乘法循环群, 它们的阶都为素数 p, P 是 G 的生成元, $e:G\times G\rightarrow G_T$ 是一个双线性映射, $h:G\rightarrow Z_p,h_1,h_2:\{0,1\}^*\rightarrow Z_p$, $H,H_1:\{0,1\}^*\rightarrow G$ 以及 $H_2:G_T\rightarrow\{0,1\}^{(\lg p)}$ 是安全的哈希函数, Z_p 为整数关于模 p 的加法群, E/D 是一种安全的对称加/解密算法 (如 CTR[AES]), Sig 是一种安全的签名算法 (如 BLS 签名), $F_1:Z_p\times Z_p\rightarrow Z_p$ 以及 $F_2:Z_p\times\{0,1\}^{poly(\ell)}\rightarrow\{0,1\}^{poly(\ell)}$ 是伪随机函数, w 为数据检索关键词, 与外包存储数据的内容相关, $poly(\ell)$ 为一个关于 ℓ 的多项式, 用于表示 w 的长度, t 是门限值, n 是 App 的总数.

$MasterKeyGen(\ell,n,t)\rightarrow(s,PS,s_i,PS_i)$. 作为分布式服务器的 App 运行执行分布式秘密共享协议 (算法 1).

算法 1. 分布式秘密共享协议.

参数说明: ℓ: 安全参数; $\{1,\ldots,n\}$: App 的索引; t: 门限值; i: 执行该算法的 App 的索引.

确保: 所有 App 共享一个服务器端密钥 s 和相应的公钥 PS; 对于 $i=1,2,\ldots,n$, S_i 计算子密钥 s_i 和相应的子公钥 PS_i.

1. 每一个 $S_i(i=1,2,\ldots,n)$ 随机选取参数 $a_{i,0},a_{i,1},\ldots,a_{i,t-1}\in Z_p^*$ 构造一个 $t-1$ 阶多项式 $f_i(x)=a_{i,0}+a_{i,1}x+\ldots+a_{i,t-1}x^{t-1}\in Z_p$, 使得 $f_i(0)=a_{i,0}$;

2. S_i 计算 $a_{i,0}P$、$a_{i,\tau}P(\tau=1,2,\ldots,t-1)$ 并公开, S_i 计算 $f_i(j)$ 并通过安全信道发送给 $S_j(j=1,2,\ldots,n;j\neq i)$;

3. S_i 收到 $f_j(i)$ 后验证等式 $f_j(i)P=\sum_{\gamma=0}^{t-1}i^{\gamma}\cdot a_{j,\gamma}P$ 是否成立; 若验证通过则接受 $f_j(i)$; 否则拒绝;

4. S_i 的子密钥 $s_i=\sum_{\gamma=1}^{n}f_{\gamma}(i)$, 其对应的子公钥 $PS_i=s_iP$. 在 App $\{S_1,S_2,\ldots,S_n\}$ 间共享的私钥 $s=\sum_{i=1}^{n}a_{i,0}$, 相应公钥 $PS=\sum_{i=1}^{n}a_{i,0}P$.

各 App (后文中用 A_i 表示, 其中 $i=1,2,\ldots,n$) 执行分布式秘密共享协议, 得到其密钥共享 $k_i=s_i$, 相应公钥 $PK_i=PS_i$, App 端公钥为 $PK=PS$.

$Sign(x,r)\rightarrow\sigma$: 签名算法. 用户 U 生成用户名 ID_u、不可复用的长效口令 psw_u. U 均匀一致地选取参数 $r\in Z_p^*$, 如公式 (1) 计算盲化口令并将 (ID_u,psw_u^*) 发送给 A_i $(i=1,2,\ldots,n)$.

$$psw_u^*=r\cdot H(psw_u) \tag{1}$$

对于 $i\in[1,n]$, A_i 检查本地存储中 ID_u 是否已存在; 若已存在, 返回 U 身份重复; 否则, A_i 存储 ID_u, 如公式 (2) 计算其对盲化口令的签名并发送给 U. U 检查等式 $e(\sigma_i^*,P)=e(psw_u^*,PK_i)$ 是否成立; 若等式成立, 则 σ_i^* 有效; 否则, U 拒绝 σ_i^*.

$$Sig(psw_u^*)=\sigma_i^*=k_i\cdot psw_u^* \tag{2}$$

$Encap(\sigma_1,\sigma_2,\ldots,\sigma_t)\rightarrow\sigma$: 签名封装算法 (算法 2).

算法 2. 签名封装算法.

参数说明: ℓ: 安全参数; $\{1,\ldots,n\}$: 服务器的索引; t: 门限值.

确保: 收到 t 个来自服务器的有效签名 $\{\sigma_1,\sigma_2,\ldots,\sigma_t\}$ 后, U 可计算签名 σ.

1. U 计算 $w_\eta=\prod_{1\leqslant\iota\leqslant t,\iota\neq n}\frac{\iota}{\iota-\eta}$;
2. U 计算封装签名 $\sigma=\sum_{\eta=1}^{t}w_\eta\sigma_\eta$

收到 t 个有效的 σ_i^* 后 (用 $\{\sigma_1^*,\sigma_2^*,\ldots,\sigma_t^*\}$ 表示), U 计算封装后的盲化口令签名 σ_u^*, 根据公式 (3) 去盲化后得到封装后的口令签名.

$$\sigma_u=r^{-1}\cdot\sigma_u^* \tag{3}$$

$RetriSerPsw(\sigma_u,psw_u)\rightarrow sp_u$: App 派生密钥生成算法. U 检查等式 $e(\sigma_u,P)=e(H(psw_u),PK)$ 是否成立; 若等式不成立, 则 σ_u 计算错误; 否则, 根据盲化口令的签名, U 根据公式 (4) 计算 App 派生密钥.

$$sp_u=F_1(h_1(\sigma_u),psw_u) \tag{4}$$

对于 $i=1,2,\ldots,n$, U 根据公式 (5) 计算凭证发送给 A_i, A_i 存储 sp_i 并对应 ID_u, 完成用户注册.

$$sp_i=h_2(sp_u\|i) \tag{5}$$

用户完成注册后, 为便于后续通过设备在 App 进行安全便捷的登录与使用, 用户需要对设备进行授权.

$Auth(sp_i,K_D,K_R)\rightarrow Aut_i$: 设备授权算法. U 生成两对安全的短效公私钥对 (K_D,S_D), (K_R,S_R) 并存储在待授权设备 D 上, 随后, U 均匀一致地选取参数 $\alpha\in Z_p^*$, 如公式 (6) 计算盲化口令并将 (ID_u,psw_u') 发送给 A_i $(i=1,2,\ldots,n)$.

$$psw_u'=\alpha\cdot H(psw_u) \tag{6}$$

对于 $i=1,2,\ldots,n$, A_i 根据公式 (7) 计算其对盲化口令的签名, 随机选取 $r_i\in Z_p$, 将 $\{\sigma_i',r_i\}$ 发送给 U.

$$Sig(psw_u')=\sigma_i'=k_i\cdot psw_u' \tag{7}$$

U 获取 r_i 和 σ_i', 检查等式 $e(\sigma_i',P)=e(psw_u',PK_i)$ 是否成立; 若等式成立, 则 σ_i' 有效; 否则, U 拒绝 σ_i'. 收到 t 个有效的 σ_i', 即 $\{\sigma_1',\sigma_2',\ldots,\sigma_t'\}$ 后, U 计算封装后的盲化口令签名 σ_u', 根据公式 (8) 去盲化后得到封装后的口令签名.

$$\sigma_u=\alpha^{-1}\cdot\sigma_u' \tag{8}$$

U 检查等式 $e(\sigma_u,P)=e(H(psw_u),PK)$ 是否成立; 若等式不成立, 则 σ_u 计算错误; 否则, 根据盲化口令的签名, U 根据公式 (4) 计算 App 派生密钥 sp_u. 对于 $i=1,2,\ldots,n$, U 根据公式 (5) 计算凭证 sp_i 并发送 $E_{sp_i}(r_i,K_D,K_R)$ 给 A_i, A_i 解密并获取 r_i, 验证 r_i 是否正确; 若解密失败或 r_i 错误, A_i 拒绝该操作; 否则, A_i 根据公式 (9)、(10) 计算证书签名并将 $EAutToken_i$ 发送给 U, $Token$ 中包含 K_D、K_R、U 的信息、过期时间以及其他辅助信息.

$$Aut_i=k_i\cdot H(Token) \tag{9}$$

$$EAutToken_i = E_{sp_i}(Aut_i \| Token) \tag{10}$$

U 计算 $D(EAutToken_i)$ 获得 Aut_i 和 $Token$, 验证等式 $e(Aut_i, P) = e(H(Token), PK_i)$ 是否成立; 若等式成立, 则 Aut_i 有效; 否则, U 拒绝该操作. 收到 t 个有效证书签名 (用 $\{Aut_1, Aut_2, \ldots, Aut_t\}$ 表示) 后, U 计算封装后的证书签名 Aut_u, 检查等式 $e(Aut_u, P) = e(H(Token), PK)$ 是否成立; 若等式不成立则协议终止; 否则, U 接受 Aut_u, 如公式 (11) 计算设备证书.

$$Cert_D = \{Token, Aut_u\} \tag{11}$$

设备授权成功 (获取设备证书) 后, 用户可以通过授权设备在 App 和云存储服务器进行登录和使用. 用户在云存储服务器上的登录与其在 App 的登录步骤类似, 出于简洁性的考量, 以下只具体展示用户在 $A_i (i = 1,2,\ldots,n)$ 登录的具体步骤.

U 使用设备 D 通过客户端联系 A_i 的后端服务器进行登录请求, A_i 为客户端生成唯一的会话标识 ID_S, D 获取客户端的 ID_S 后生成一个随机的账户 ID 字符串 ID_a 以及一个短效的账户密钥对 (K_a, S_a). 随后, D 创建一个证书签名请求 (certificate signing request, CSR), 其中包含 $Sig_{S_a}(ID_a, K_a)$, D 计算 $Sig_{S_D}(CSR)$ 并将 CSR、$Sig_{S_D}(CSR)$ 及 $Cert_D$ 发送给 A_i.

A_i 验证 D 是否有权获取 $Cert_D$ 中列出的用户名的证书, 并同时拥有 S_D 和 S_a; 若验证失败, 则 A_i 拒绝该操作; 否则, A_i 将 ID_a 与该用户名关联起来, 并返回一个账户证书 $Sig_{k_i}(Cert_a)$, $Cert_a$ 中列出了 ID_a 和 K_a.

若为首次登录, D 生成用于与 A_i 会话的短效密钥对 (K_S, S_S), 生成会话证书 $Cert_S$, 证书包含 ID_S、A_i 的域名以及 K_S, D 向 A_i 发送 $Cert_a$, $Sig_{S_a}(Cert_S)$ 和 $Sig_{S_S}(ID_S)$ 请求登录. A_i 将 K_S 与 U 的账户关联, 随后只允许来自证明其拥有 S_S 的 D 登录.

若不为首次登录, D 需查找其用于 A_i 的会话密钥对 (K_S, S_S), 通过计算 $Sig_{S_S}(ID_S)$ 来证明其拥有 S_S 以完成登录.

用户数据的外包存储由发送数据的 App (A_S, $A_S \in \{A_1, A_2, \ldots, A_n\}$) 完成.

首先, U 为其授权设备 D 从 Z_p^* 中随机选取用于陷门计算的私钥 α, 如公式 (12) 计算对应的公钥, 给定 U 的使用数据的数据文件 data 及其关键词 w, A_S 计算 $E_{K_R}(\text{data})$ 并发送至云存储服务器 C.

$$Q_D = \alpha P \in G \tag{12}$$

$RetriSerKw(\sigma_w, w) \to sd_w$: 强化关键词算法. $A_i (i = 1,2,\ldots,n)$ 执行分布式秘密共享协议得到其子密钥 $sk_i = s_i$, 子公钥 $PSK_i = PS_i$, 计算得到相应的公钥 $PKS = PS$. A_S 均匀一致地选择 $\beta \in Z_p^*$, 计算 $w' = \beta \cdot H(w)$, 由于 A_S 在 A_i $(i = 1,2,\ldots,n)$ 中, 故 A_S 可以根据公式 (13) 使用其子密钥计算一个对 w' 的签名.

$$Sig_{sk_S}(w') = \sigma_S = sk_S \cdot w' \tag{13}$$

随后, A_S 随机选择其余 $t-1$ 个 App 的集合 $T = \{A_{i_1}, A_{i_2}, \ldots, A_{i_{t-1}}\}$, $A_S \notin T$, 将 w' 发送给 T. 收到 w' 后, A_k $(k = i_1, i_2, \ldots, i_{t-1})$ 根据公式 (14) 计算其对 w' 的签名并将 σ_k 发送给 A_S, A_S 验证等式 $e(\sigma_k, P) = e(w', PSK_k)$ 是否成立; 若验证失败, A_S 拒绝 σ_k; 否则, A_S 在本地存储 σ_k.

$$Sig_{sk_k}(w') = \sigma_k = sk_k \cdot w' \tag{14}$$

A_S 共需要接收到 t 个有效签名 (用 $\{\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_{t-1}}\}$ 表示), 使用 t 个有效签名 $\{\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_{t-1}}, \sigma_S\}$ 计算盲化后的封装签名 σ_w^*, 如公式 (15) 去盲化后得到封装签名.

$$\sigma_w = \beta^{-1} \cdot \sigma_w^* \tag{15}$$

A_S 计算验证等式 $e(\sigma_w, P) = e(H(w), PSK)$ 是否成立; 若等式不成立, 则 σ_w 计算错误; 否则, A_S 可根据公式 (16) 计算强化关键词.

$$sd_w = F_2(h(\sigma_w), w) \tag{16}$$

A_S 随机选取 $\chi \in Z_p^*$, 根据公式 (17)、(18) 计算强化关键词的密文并将 C_{sd_w} 发送给 C.

$$\tau = e(H_1(sd_w), \chi Q_D) \tag{17}$$

$$C_{sd_w} = (\chi P, H_2(\tau)) \tag{18}$$

C 生成 data 的密文 C_{data} 与 C_{sd_w} 相对应的索引表如表 1 所示.

表 1 数据-关键词索引表

强化关键词密文	数据密文	数据存储地址
$C_{sd_w}1$	$C_{\text{data}1},\ldots,C_{\text{data}p}$	$addr_1,\ldots,addr_p$
$C_{sd_w}2$	$C_{\text{data}2},\ldots,C_{\text{data}q}$	$addr_2,\ldots,addr_q$
…	…	…
$C_{sd_w}n$	$C_{\text{data}n},\ldots,C_{\text{data}m}$	$addr_n,\ldots,addr_m$

$Trapdoor(w)\rightarrow td_w$: 陷门计算算法. 在数据请求阶段, 请求数据流转的 App (用 A_R 表示, $A_R\in\{A_1,A_2,\ldots,A_n\}$) 产生用于数据请求的公私钥对 (K_d,S_d), A_R 向 U 的登录设备 D 发送包含 w 和 K_d 的数据请求 $E_{K_R}(quest)$, D 计算 $D_{S_R}(quest)$ 获得 w, 随后与 t 个 App 交互获取 sd_w, 这一过程与数据外包存储阶段中 A_S 执行的过程相同. D 根据公式 (19) 计算陷门并发送给 C.

$$td_w = \alpha H_1(sd_w) \tag{19}$$

$Test(td_w)\rightarrow C_{\text{data}}$: 数据检索算法. C 获取 td_w 后验证等式 $H_2(e(td_w,\chi P)) = H_2(\tau)$ 是否成立; 若等式不成立, 拒绝该操作; 否则, C 将 C_{sd_w} 对应的所有 C_{data} 发送给 D.

U 计算 $D_{S_R}(C_{\text{data}})$ 后对其中允许进行共享的 data′ 计算 $Sig_{S_S}(E_{K_d}(\text{data}'))$ 发送给 A_R, A_R 解密后获取数据.

3.2 正确性证明

为了证明方案的正确性, 需要从以下两方面完成证明.

首先, 需证明用户只需要安全存储长效口令, 即可通过 App 的身份验证.

在登录阶段, 用于证明用户身份的用户凭证为 $sp_i = h_2(sp_u\|i)$, 其中 sp_u 为 App 派生密钥, 具体地, $sp_u = F_1(h_1(\sigma_u), psw_u)$, 其中 $\sigma_u = kH(psw_u)$ 本质上是一个 BLS 签名[55], 即 σ_u 是一个确定性签名. 因此, 只要 U 拥有正确的 psw_u, 就可以计算出确定性的 App 派生密钥 sp_u, 进而得到其用户凭证 sp_i 来通过 App 的身份认证.

其次, 需证明对于不同的 App 集合 (即 $T=\{A_{i_1},A_{i_2},\ldots,A_{i_t}\}, A_S\notin T$) 的选择, 相同关键词的强化关键词相同, 这依赖于签名封装算法的正确性.

具体地, 为简洁起见, 假设 t 个 App 产生了 t 个有效签名 $\{\sigma_1,\sigma_2,\ldots,\sigma_t\}$, 则 σ_w 可以表示为如下形式:

$$\begin{aligned}\sigma_w &= r^{-1}\sum_{i=1}^{t} w_i\sigma_i \\ &= r^{-1}\left(\sum_{i=1}^{t} s_i \prod_{1\leqslant j\leqslant t, j\neq i}\frac{j}{j-i}\right)\cdot w' \qquad (20)\\ &= r^{-1}\cdot r\cdot s\cdot H(w) \qquad (21)\\ &= s\cdot H(w)\end{aligned}$$

基于拉格朗日插值, 公式 (20) 和 (21) 成立.

4 安全性分析

本节形式化地证明了 CADC 对半诚实的 App 和云存储服务器的可靠性. 首先, 引出定理 1–定理 3 以及引理 1、引理 2, 并完成证明. 其次, 基于定义 1, 在定理 4 中描述了 CADC 的可靠性. 根据定理 1–定理 3 及引理 1、引理 2 的正确性, 可以完成对定理 4 的证明. 具体的定义及证明过程如下.

定义 2. 遵循现有工作中针对选择关键词猜测攻击 (SS-CKGA) 的语义安全的安全定义[53], SS-CKGA 安全游戏由一个敌手 A 与挑战者 B 之间进行的交互挑战定义如下.

1) 初始化: 环境 ε 初始化该方案, 向 B 发送 $pk_R(Q_D), sk_R(\alpha), Q(PKS)$, 向 A 发送 pk_R, Q.

2) 问答阶段 1: A 可以适应性地向 B 请求任意关键词 w 的陷门或强化关键词密文.

3) 挑战阶段: A 向 B 发送两个先前未请求过的关键词 w_0, w_1, B 随机选取 $b \in \{0,1\}$ 并生成 $sd_{w_b}, C_{sd_{w_b}}, td_{w_b}$, B 发送 $C_{sd_{w_b}}, td_{w_b}$ 给 A.

4) 问答阶段 2: A 可以继续适应性地向 B 请求除 w_0, w_1 以外任意关键词 w 的陷门或强化关键词密文.

5) 输出阶段: A 输出猜测结果 $b' \in \{0,1\}$, 若 $b' = b$ 则 A 获胜; 否则, A 失败.

A 赢得 SS-CKGA 挑战的优势定义为: $Adv_{\mathrm{A}}^{\text{SS-CKGA}}(\ell) = |\Pr[b' = b] - 1/2|$.

根据以下两个定理, CADC 可以被证明是 SS-CKGA 安全的.

定理 1. 如果 PPT 敌手 A 可以破坏 CADC 的 SS-CKGA 安全, 那么一定存在一个 PPT 敌手 B 可以破坏底层签名 (BLS 签名[55]) 的安全性.

证明: 定理 2 的证明同服务器协助的 PEKS 方案[53]对 SS-CKGA 安全的证明. 由于 B 破坏底层签名安全性的概率是可忽略的, 故 A 在 SS-CKGA 中获胜的优势是可忽略的, 即 $Adv_{\mathrm{A}}^{\text{SS-CKGA}}(\ell) = negl(\ell)$.

定理 2. 假设 Shamir 秘密共享方案[56]是安全的, 那么 CADC 中以分布式的方式生成给定关键词的签名的过程就不会破坏 CADC 的 SS-CKGA 安全.

证明: 在 CADC 中, w 的签名 σ_w 由多个作为密钥服务器的 App 参与生成, 其中各 App 使用 Shamir 秘密共享方案共享私钥 s, 并使用 s 对 w 进行签名. 但对于敌手 A 来说, 分布式方式 (即多个密钥服务器的情况) 和集中式方式 (即单个密钥服务器的情况) 并没有区别. 因为在问答阶段 1 和问答阶段 2 中, 当 A 向 B 请求强化关键词密文或陷门时, B 与签名预言机交互, 接下来的挑战都基于该预言机的输出. 在挑战阶段, B 不能与签名预言机交互, 而是随机选择一个元素作为 w_b 的签名以生成挑战关键词密文和陷门. 实际上, 上述所有过程对敌手 A 都是透明的, 不会影响 A 在 SS-CKGA 挑战中获胜的优势.

定义 3. 针对选择关键词攻击的不可区分性 (IND-CKA) 挑战定义如下.

1) 初始化: 根据安全参数 ℓ, 挑战者 B 产生 App 端秘密共享, 其中一个 App (不失一般性地, A_1) 的公共参数 sk_1, PSK_1 并发送给敌手 A.

2) 挑战阶段: A 向 B 发送两个关键词 w_0, w_1, B 随机选取 $b \in \{0,1\}, r \in Z_p$, 计算签名 σ_1 并发送给 A.

3) 输出阶段: A 输出猜测结果 $b' \in \{0,1\}$, 若 $b' = b$, A 获胜.

A 赢得 IND-CKA 挑战的优势被定义为 $Adv_{\mathrm{A}}^{\text{IND-CKA}}(\ell) = |\Pr[b' = b] - 1/2|$.

根据以下定理, 方案是 IND-CKA 安全的.

定理 3. 如果存在一个 PPT 敌手 A 可以破坏 CADC 的 IND-CKA 安全, 那么就存在一个 PPT 敌手 B 可以破坏门限盲签名方案[57]的盲性.

证明: 为了证明定理 4, 不失一般性地, 假设敌手 A 为 A_1, B 将 A 作为子程序运行, 且对于挑战者来说是透明的.

在初始化时, B 收到来自挑战者的 sk_1, PSK_1 并发送给 A. 在挑战阶段, B 收到来自 A 的 w_0, w_1 并发送给挑战者, 挑战者随机选取 $b \in \{0,1\}, r \in Z_p$, 计算 $w_b' = rH(w_b)$ 并发送给 B, B 计算签名 $\sigma_1 = sk_1 w_b'$ 并发送给 A. 在输出阶段, A 向 B 发送猜测 b', B 也输出 b' 作为其对 b 的猜测.

在上述过程中, 对于挑战者, B 是破坏了门限盲签名方案[57]的盲性的敌手; 对于 A, B 是 CADC 的 IND-CKA 挑战中的挑战者. 因此, B 破坏门限盲签名方案[57]的盲性的优势大于等于 A 在 CADC 挑战中获胜的优势 $Adv_{\mathrm{A}}^{\text{IND-CKA}}(\ell)$. 若 $Adv_{\mathrm{A}}^{\text{IND-CKA}}(\ell) > negl(\ell)$, 则 B 一定能够以不可忽略的优势破坏门限盲签名方案[57]的盲性.

此外, 在方案中即使部分 App 的子密钥泄露, 方案的安全性仍旧可以得到保证.

密钥具有如下形式:

$$sk = \sum_{i=1}^{t} w_i sk_i = \sum_{i=1}^{t} \left(\prod_{\substack{1 \leqslant \eta \leqslant t \\ \eta \neq i}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) \right) \tag{22}$$

由于敌手无法获取对于 $j = 1,2,\ldots,n$, $i = 1,\ldots,t$ 的全部 $f_j(i)$ 值, 因此敌手无法恢复出密钥 sk.

假设敌手 A 可以破坏任意 t' ($t' < t$) 个作为身份服务器的 App 并获取其秘密共享. A 的目标是获取受害用户

U' 的长效口令. 注意到第 i 个 App 的用户凭证为 $sp_i = h_2(sp_u||i)$, 由 App 派生密钥 $sp_{u'}$ 计算得到. 由于哈希函数的抗第一原象性, 对于 $i \in [1,n]$, 给定 sp_i, 计算 $sp_{u'}$ 是计算不可行的. A 破解 $sp_{u'}$ 的最高效的方法是破解 $sp_{u'}$ 的生成过程. $sp_{u'}$ 的安全性由引理 1 和引理 2 保证, 确保了 CADC 能够抵御敌手 A.

引理 1 (不可预测性). 在 CADC 中, App 派生密钥是不可预测的. 这意味着, 给定用户 U' 的口令 $psw_{U'}$, 即使敌手可以破坏 $t'(t' < t)$ 个 App, 也无法预测相应的 App 派生密钥 $sp_{U'}$.

证明: 定义 CADC 的不可预测性挑战如下.

1) 环境 ε 初始化该方案, 生成一个服务器端密钥 k, 使用 (t,n) 门限秘密共享协议生成 n 个服务器端密钥共享 $\{k_1,k_2,\ldots,k_n\}$ 并产生相应的公共参数 PP 发送给模拟器 S, S 将 PP 转发给 A.

2) A 选取 t' 个索引 $T = \{i_1,i_2,\ldots,i_{t'}\}$ 发送给 S, S 转发给 ε. ε 回复 $\left\{k_{i_1},k_{i_2},\ldots,k_{i_{t'}}\right\}$ 给 S, S 转发给 A 并设置 q_1, $q_2,\ldots,q_n = 0$.

3) A 随机选取 $x \in Z_p^*, r \in Z_p^*$ 并计算 $c = r \cdot H(x)$.

4) 对 $i \in [1,n]/T$, A 向 S 请求 $\sigma_i = k_i c$.

5) 收到请求后, S 设置 $q_i = q_i + 1$ 并将请求转发给 ε.

6) ε 计算 σ_i 并发送给 S, S 将 σ_i 转发给 A.

7) A 对不同的 i 重复上述请求, 最终输出 σ.

当且仅当 $\sum_{i=1}^{n} q_i < t - t'$, 且 σ 是对 c 的有效签名时, A 在不可预测性挑战中获胜.

定义 A 在不可预测性挑战中获胜的概率为 $Adv_{\mathrm{A}}^{\mathrm{predi}}(\ell) = \Pr\left[(\sigma = \sigma_c) \cap \left(\sum_{i=1}^{n} q_i < t - t'\right)\right]$.

可证明如果 A 能够以一个不可忽略的概率赢得不可预测性挑战, 那么 S 也能够以同样的概率破坏 BLS 签名的存在不可伪造性. 具体地, 在交互过程中, S 可以获取 $\{k_\iota,\sigma_\iota\}_{\iota=T'}$, $\{\sigma_i\}_{i\in[1,n]/T'}$ 和 σ. 注意到无论 A 采取何种策略, 都不会比请求次数满足 $\sum_{i=1}^{n} q_i + t' = t - 1$ 时更有效, 因此, 不妨假设 S 可以收集到 $t-1$ 个服务器端密钥共享下的 $t-1$ 个签名 $\sigma_{i'_1},\sigma_{i'_2},\ldots,\sigma_{i'_{t-1}}$ 以及一个服务器端密钥下的签名 (由 A 输出). 假设 A 的目标是在未知第 t 个服务器端密钥共享的情况下伪造 c 的签名, 则 S 可根据公式 (23) 伪造 BLS 签名, 其中 $w_\eta = \prod_{i'_1 \leqslant \iota \leqslant i'_{t-1}, \iota \neq \eta} \frac{\iota}{\iota - \eta}$, $w_t = \prod_{i'_1 \leqslant \iota \leqslant i'_{t-1}} \frac{\iota}{\iota - t}$ 输出.

$$\sigma_t = \left(\sigma - \left(\sum_{\eta=i'_1}^{i'_{t-1}} w_\eta \sigma_\eta\right)\right) \cdot \frac{1}{w_t} \tag{23}$$

引理 2 (不可见性). 在本方案中, App 派生密钥是不可见的. 这意味着, 给定用户 U' App 派生密钥 $sp_{U'}$, 敌手无法获取相应口令 $psw_{U'}$ 的任何信息 (如 $psw_{U'}$ 的哈希值).

证明: 由于 r 的随机性, 敌手只能收集到不同口令盲化后的信息. 定义一个不可见性证明过程如下.

1) 环境 ε 初始化该方案, 生成一个 App 端密钥 k, 使用 (t,n) 门限秘密共享协议生成 n 个 App 端密钥共享 $\{k_1,k_2,\ldots,k_n\}$ 并产生相应的公共参数 PP 发送给模拟器 S, S 将 PP 转发给 A.

2) A 随机选取 $c \in Z_p$ 发送给 S, S 转发给 ε.

3) ε 随机选取 r 并计算 $\sigma_c = rk \cdot H(c)$, 将 App 派生密钥 $sp_c = F_1(h_1(\sigma_c),c)$ 发送给 S, S 收到后转发给 A.

4) A 可重复进行步骤 2)、3), 次数上限由安全参数 ℓ 的多项式 $poly(\ell)$ 界定.

5) A 随机选取 $c_0,c_1 \in Z_p$ 并发送给 S, S 收到后转发给 ε.

6) ε 随机选取 $b \in \{0,1\}, r' \in Z_P^*$. 若 $b = 0$, ε 计算 $sp_{c_b} = F_1(h_1(kr'H(c_b)),c_b)$; 否则 ε 随机选取一个函数 $F' \in Func_{Z_p}$, 计算 $sp_{c_b} = F'(c_b)$. ε 将 sp_{c_b} 发送给 S, S 转发给 A.

7) A 输出 b'.

当且仅当 $b' = b$ 时, A 在不可见性证明过程中获胜.

A 在不可见性证明过程中获胜的概率为 $\Pr_{\mathrm{A}}^{\mathrm{Obli},\ell} = |\Pr[b' = b] - 1/2|$.

可证明如果 A 能够以一个不可忽略的概率在不可见性证明过程中获胜, 那么 S 也能够以同样的概率 $\Pr_{\mathrm{A}}^{\mathrm{Obli},\ell}$ 破

坏伪随机函数 F 的安全性. 具体地, 在交互过程中, A 的目标是在未知 App 端密钥的情况下区分 App 派生密钥 sp_{c_b}, 则 S 能够输出同样的 b' 作为其对伪随机函数 $F'(c_b)$ 与 $F_1(h_1(kr'H(c_b)),c_b)$ 的区分结果, 因此, S 区分伪随机函数 F 的优势 $\mathrm{Pr}_S^{\mathrm{Obli},\ell}=\mathrm{Pr}_{\mathrm{A}}^{\mathrm{Obli},\ell}$. 由于伪随机函数的安全性, $\mathrm{Pr}_{\mathrm{A}}^{\mathrm{Obli},\ell}=\mathrm{Pr}_S^{\mathrm{Obli},\ell}=negl(\ell)$, A 在不可见性证明过程中获胜的概率是可忽略的.

基于定义 1, 定理 4 描述了 CADC 的可靠性.

定理 4. 假设 CADC 的签名算法都满足存在不可伪造性, 对称加密算法和公钥加密算法是 CPA 安全的, 哈希函数都是抗碰撞的且合谋的半诚实 App 数量小于门限值 t, 则对于任意 PPT 敌手 A, 在安全参数 ℓ 下, A 在 CADCsound 游戏中获证的概率都是可忽略的, 即 $\mathrm{Pr}[CADCsound_{\mathrm{A}}(\ell)=1]=negl(\ell)$.

证明: 根据上述定理 2–定理 4, 以及引理 1、引理 2. 对于任意 PPT 敌手 A, 在安全参数 ℓ 下, A 在 CADCsound 游戏中获证的概率满足公式 (24), 因此, A 在 CADCsound 游戏中获证的概率是可忽略的, CADC 的可靠性证明完成.

$$\mathrm{Pr}[CADCsound_{\mathrm{A}}(\ell)=1]=Adv_{\mathrm{A}}^{\text{SS-CKGA}}(\ell)+Adv_{\mathrm{A}}^{\text{IND-CKA}}(\ell)+Adv_{\mathrm{A}}^{\text{predi}}(\ell)+\mathrm{Pr}_{\mathrm{A}}^{\mathrm{Obli},\ell}=negl(\ell) \tag{24}$$

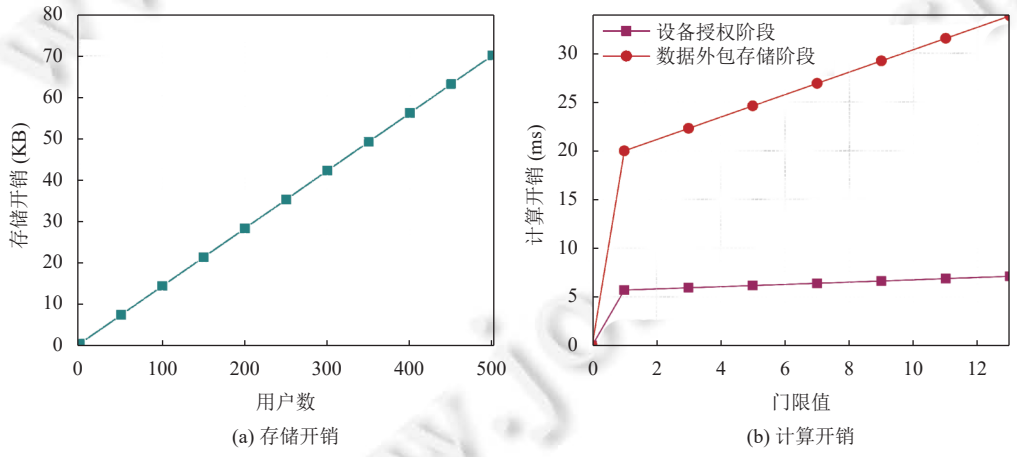
5 性能分析

CADC 的性能通过仿真实验进行评估分析. 仿真实验使用 PBC 库 (https://crypto.stanford.edu/pbc/manual/) 实现, 运行在配置 Windows 10 操作系统、Intel Core i7-10510U CPU 和 16 GB 机带 RAM 环境的计算机上. 根据双线性配对曲线 Type A, 选定安全参数 $\ell=160$ 比特 (其安全保证相当于具有 160 位密钥的 AES).

5.1 存储开销

在 CADC 中, 用户只需维护其用户名和长效口令以用于向 App 请求身份验证和认证证书, 无需额外存储任何信息. 用户已授权的设备需要存储短效公私钥对、设备证书和会话密钥, 其存储开销固定且不足 1 KB.

本节主要关注 CADC 在 App 端的存储开销. 除系统公共参数外, App 还需要存储每个用户的 3 组主密钥共享和服务器端密钥共享、用户名和凭证. 图 3(a) 描述了 App 端的存储开销, 当用户数达到 500 时, App 端的存储开销不超过 80 KB. 由此可见, CADC 可以为用户提供便捷高效的部署, 且在 App 端的存储开销极低.



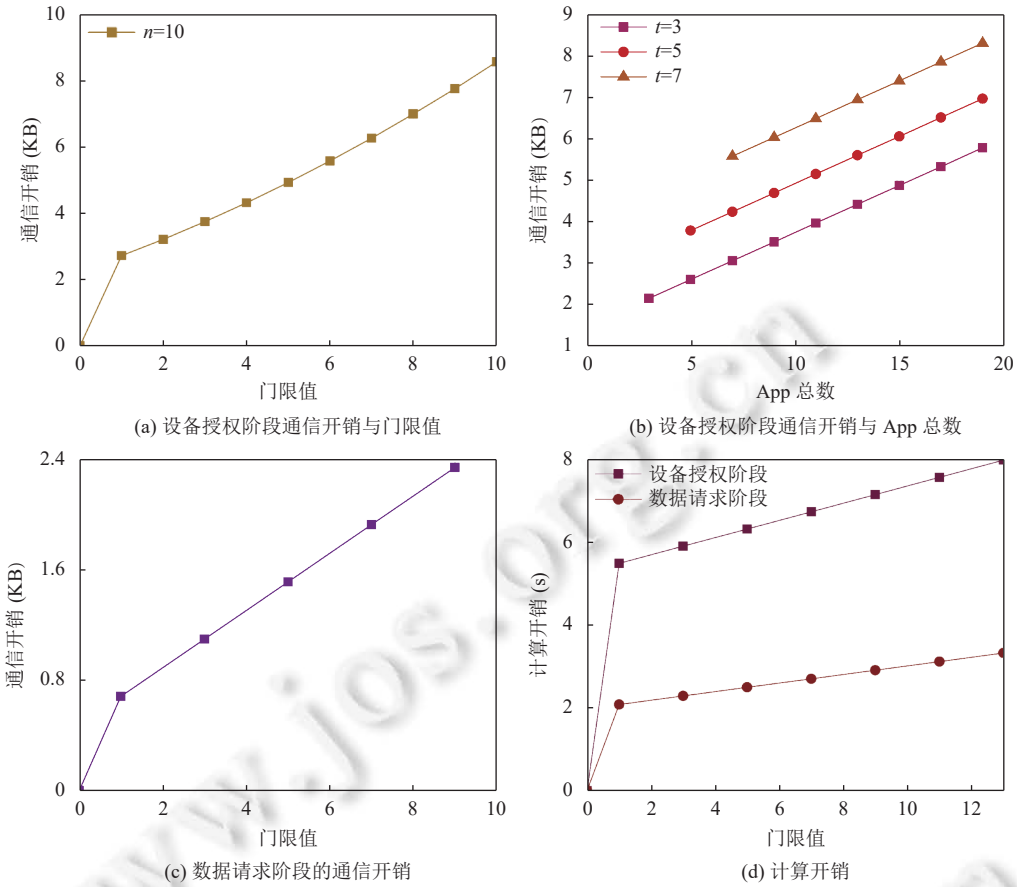
(a) 存储开销 (b) 计算开销

图 3 App 端开销

5.2 通信开销

注册阶段在 CADC 中可看作一次性操作, 因此不分析此阶段的通信开销.

设置 Token 的长度为 256 字节, 图 4(a) 和图 4(b) 分别描述了设备授权阶段用户端的通信开销与门限值和 App 总数的关系.



(a) 设备授权阶段通信开销与门限值　(b) 设备授权阶段通信开销与 App 总数

(c) 数据请求阶段的通信开销　(d) 计算开销

图 4　用户端开销

在登录阶段, 用户与 App 间需要进行 4 轮通信. 设置设备和账户的公私钥的长度均为 1 024 比特 (根据 RSA 算法的安全性要求), Token 的长度设置为 256 字节, 此阶段的通信开销固定为 1.2 KB.

在数据请求阶段, 用户需要与 App、云存储服务器通信以完成数据检索与传输, 设置外包存储的数据文件大小为 256 字节, 图 4(c) 描述了数据请求阶段用户端的通信开销.

由以上实验结果可以看出, CADC 在用户端的通信开销极低, 不会对方案的实际应用部署造成阻碍. 此外, 在 CADC 中, 作为服务器的 App 可以很好地配备并保持在线状态, App 端的通信成本同样是可接受的, 不会成为 CADC 方案应用的瓶颈.

5.3　计算开销

表 2 中总结了在 CADC 中用到的基本密码学函数.

表 2　基本密码学函数表

参数	含义	参数	含义
$Pair_{G_T}$	双线性映射的计算	Add_G	群 G 中的加法运算
$Hash_{Z_p}$	将一个值哈希到群 Z_p 中	Mul_G	群 G 中的乘法运算
$Hash_G$	将一个值哈希到群 G 中	Exp_{Z_p}	群 Z_p 中的指数运算
C_F	计算伪随机函数的值	Enc/Dec	对称加密/解密算法
Add_{Z_p}	群 Z_p 中的加法运算	Sig/Ver	数字签名/签名验证算法
Mul_{Z_p}	群 Z_p 中的乘法运算	E/D	公钥加密/解密算法

CADC 各阶段的计算开销如表 3 所示.

表 3 计算开销

阶段	用户端	App端
设备授权阶段	$(2t+6)\cdot Pair_{G_T}+(2t+2)\cdot Mul_G+(2t-2)\cdot Add_G+3Hash_G+(2t-2)\cdot Mul_{Z_p}+(n+1)\cdot Hash_{Z_p}+n\cdot Enc+n\cdot Dec+Exp_{Z_p}+C_F$	$(nt+n+1)\cdot Mul_G+(t-1)(n-1)\cdot Mul_{Z_p}+(nt-t+n-2)\cdot Add_{Z_p}+t(n-1)\cdot Add_G+Enc+Dec+Hash_G$
登录阶段	首次登录: $2\cdot Exp_{Z_p}+4\cdot Sig$ 非首次登录: $Exp_{Z_p}+3\cdot Sig$	$4Ver$
数据外包存储阶段	0	$(t+4)\cdot Pair_{G_T}+(nt+t+n+3)\cdot Mul_G+n(t-1)\cdot Mul_{Z_p}+(nt-t+n-2)\cdot Add_{Z_p}+(nt-1)\cdot Add_G+3Hash_G+2Hash_{Z_p}+Enc+C_F$
数据请求阶段	$(t+3)\cdot Pair_{G_T}+3Mul_G+(t-1)\cdot Add_G+3Hash_G+(t+1)\cdot Mul_{Z_p}+Hash_{Z_p}+E+Dec+D+Sig+C_F$	$Ver+E+D$

由于注册阶段在整个方案中为一次性操作且计算开销小于设备授权阶段, 故本节不讨论注册阶段的计算开销.

对于用户端, 在设备授权阶段, 用户需使用待授权设备与各 App 交互, 计算 App 派生密钥和用于身份验证的用户凭证, 并根据用户凭证计算得到设备证书; 在登录阶段, 用户使用已授权设备进行认证登录, 用户端的计算开销固定; 在数据请求阶段, 用户需与门限个 App 交互以获取强化关键词, 计算陷门并向云存储服务器检索数据. 为便于计算, 设置 App 总数 $n=30$, 图 4(d) 描述了用户端的计算开销.

用户使用已授权设备登录 App 的计算开销大小固定且可忽略不计, 设备授权阶段的执行次数与用户设备数量和设备证书有效期有关, 在设备证书到期前, 用户无需重复进行设备授权阶段. 在实验设定数据下, 数据请求阶段的计算开销低于 3.5 s. 为保证方案安全性与便携性, 两阶段的用户端计算开销都是可接受的.

对于 App 端, 在设备授权阶段, App 需执行分布式秘密共享协议, 计算主秘密和秘密共享, 协助待授权设备获取用户凭证, 为待授权设备产生令牌 Token; 在登录阶段, App 需验证已授权设备的证书合法性, 相应的计算开销固定; 在数据外包存储阶段, 发送数据的 App 需与其余门限个 App 交互获取强化关键词, 发送强化关键词密文和对应的数据文件的密文至云存储服务器; 在数据请求阶段, 请求数据的 App 向用户已登录的授权设备发送数据请求, 由用户向云存储服务器检索数据, 相应的 App 端计算开销固定. 为便于计算, 设置 App 总数 $n=30$, 图 3(b) 描述了 App 端的计算开销.

App 为用户已授权设备执行登录的通信开销大小固定且可忽略不计, 在数据请求阶段, 由于获取强化关键词、向云存储服务器请求数据等操作都在用户端进行, 故 App 端的计算开销大小恒定且可忽略不计. 由于作为身份验证服务器和密钥服务器的 App 是分布式的, 计算开销由多个 App 共同承担, 单个 App 在设备授权阶段的通信开销低于 1 s, 在数据外包存储阶段的计算开销低于 3.5 s, 都是可接受的.

6 总 结

本文首先设计了安全高效的移动用户身份认证方法, 利用多 App 协助的门限口令加固机制和长短效秘密结合的便捷身份认证机制, 实现了适用于移动用户且便携高效的安全认证机制. 其次, 提出了安全的跨信任域数据流转方法, 利用多 App 协助的公钥可搜索加密机制, 实现了面向移动用户的抵御离线 KGA 攻击的数据安全检索机制. 最后, 将便携高效的安全认证机制与面向移动用户的抵御离线 KGA 攻击的数据安全检索机制进行整合, 形成了以用户为中心的云辅助跨 App 数据安全流转方案 CADC. 本文提供了 CADC 的形式化安全定义并进行了严格的安全性证明. 仿真实验结果表明, CADC 在用户端与 App 端具有较高的效率.

由于 CADC 在数据检索阶段对数据文件进行加解密时使用已授权设备的短效公私钥对完成, 这意味着在设备短效公私钥对到期后, 需要对数据密文进行及时更新, 在应用过程中可能会造成额外的计算和通信开销; 此外, 在 CADC 的安全模型中没有考虑在线 DGA 和在线 KGA. 在未来的工作中, 我们将继续深入研究提高数据外包系统效率, 保证数据跨应用流转安全性的方法.

References

[1] Khedker U, Sanyal A, Sathe B. Data Flow Analysis: Theory and Practice. Boca Raton: CRC Press, 2009. [doi: 10.1201/9780849332517]

[2] Pasquier TFJM, Singh J, Eyers D, Bacon J. CamFlow: Managed data-sharing for cloud services. IEEE Trans. on Cloud Computing, 2017, 5(3): 472–484. [doi: 10.1109/TCC.2015.2489211]

[3] Crooks N. Efficient data sharing across trust domains. ACM SIGMOD Record, 2023, 52(2): 36–37. [doi: 10.1145/3615952.3615962]

[4] van Kleek M, Liccardi I, Binns R, Zhao J, Weitzner DJ, Shadbolt N. Better the devil you know: Exposing the data sharing practices of smartphone Apps. In: Proc. of the 2017 CHI Conf. on Human Factors in Computing Systems. Denver: ACM, 2017. 5208–5220. [doi: 10.1145/3025453.3025556]

[5] Zhang Y, Xu CX, Li HW, Yang K, Cheng N, Shen XM. PROTECT: Efficient password-based threshold single-sign-on authentication for mobile users against perpetual leakage. IEEE Trans. on Mobile Computing, 2021, 20(6): 2297–2312. [doi: 10.1109/TMC.2020.2975792]

[6] Zhang Y, Xu CX, Ni JB, Li HW, Shen XS. Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage. IEEE Trans. on Cloud Computing, 2021, 9(4): 1335–1348. [doi: 10.1109/TCC.2019.2923222]

[7] Brandtzaeg PB, Pultier A, Moen GM. Losing control to data-hungry Apps: A mixed-methods approach to mobile App privacy. Social Science Computer Review, 2019, 37(4): 466–488. [doi: 10.1177/0894439318777706]

[8] Yu DJ, Wang JJ, Liu CF. Approach to optimal staff assignment in workflows based on collaboration patterns. Ruan Jian Xue Bao/Journal of Software, 2018, 29(11): 3340–3354 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5483.htm [doi: 10.13328/j.cnki.jos.005483]

[9] Huckvale K, Torous J, Larsen ME. Assessment of the data sharing and privacy practices of smartphone Apps for depression and smoking cessation. JAMA Network Open, 2019, 2(4): e192542. [doi: 10.1001/jamanetworkopen.2019.2542]

[10] Wang Y, Fan M, Tao JJ, Lei JY, Jin WX, Han DQ, Liu T. Compliance detection method for mobile application privacy policy statement. Ruan Jian Xue Bao/Journal of Software, 2024, 35(8): 3668–3683 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7121.htm [doi: 10.13328/j.cnki.jos.007121]

[11] Jobe W. Native Apps vs. mobile Web Apps. Int'l Journal of Interactive Mobile Technologies, 2013, 7(4): 27–32. [doi: 10.3991/ijim.v7i4.3226]

[12] Zimmeck S, Story P, Smullen D, Ravichander A, Wang ZQ, Reidenberg J, Russell NC, Sadeh N. MAPS: Scaling privacy compliance analysis to a million Apps. Proc. on Privacy Enhancing Technologies, 2019, 2019(3): 66–86. [doi: 10.2478/popets-2019-0037]

[13] Dell'Amico M, Michiardi P, Roudier Y. Password strength: An empirical analysis. In: Proc. of the 2010 IEEE INFOCOM. San Diego: IEEE, 2010. 1–9. [doi: 10.1109/INFCOM.2010.5461951]

[14] Iqbal U, Wolfe C, Nguyen C, Englehardt S, Shafiq Z. Khaleesi: Breaker of advertising and tracking request chains. In: Proc. of the 31st USENIX Security Symp. Boston: USENIX Association, 2022. 2911–2928.

[15] Wu JR, Nan YH, Xing LY, Cheng JT, Lin ZM, Zheng ZB, Yang M. Leaking the privacy of groups and more: Understanding privacy risks of cross-App content sharing in mobile ecosystem. In: Proc. of the 2024 Network and Distributed System Security Symp. (NDSS). San Diego, 2024. [doi: 10.14722/ndss.2024.24138]

[16] Wu LB, Wang J, Choo KKR, He DB. Secure key agreement and key protection for mobile device user authentication. IEEE Trans. on Information Forensics and Security, 2019, 14(2): 319–330. [doi: 10.1109/TIFS.2018.2850299]

[17] Das A, Bonneau J, Caesar M, Borisov N, Wang XF. The tangled Web of password reuse. In: Proc. of the 2014 Network and Distributed System Security. San Diego, 2014. [doi: 10.14722/ndss.2014.23357]

[18] Ma J, Yang WN, Luo M, Li NH. A study of probabilistic password models. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2014. 689–704. [doi: 10.1109/SP.2014.50]

[19] Gao G, Zhang Y, Song YQ, Li SY. PrivSSO: Practical single-sign-on authentication against subscription/access pattern leakage. IEEE Trans. on Information Forensics and Security, 2024, 19: 5075–5089. [doi: 10.1109/TIFS.2024.3392533]

[20] Song DX, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In: Proc. of the 2000 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2000. 44–55. [doi: 10.1109/SECPRI.2000.848445]

[21] Li JW, Jia CF, Liu ZL, Li J, Li M. Survey on the searchable encryption. Ruan Jian Xue Bao/Journal of Software, 2015, 26(1): 109–128 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4700.htm [doi: 10.13328/j.cnki.jos.004700]

[22] Byun JW, Rhee HS, Park HA, Lee DH. Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In:

Proc. of the 3rd VLDB Workshop on Secure Data Management. Seoul: Springer, 2006. 75–83. [doi: 10.1007/11844662_6]

[23] Zhang XL, Gu DW, Zhang C. Issues of identity verification of typical applications over mobile terminal platform. Chinese Journal of Network and Information Security, 2020, 6(6): 137–151 (in Chinese with English abstract). [doi: 10.11959/j.issn.2096-109x.2020081]

[24] Baum C, Frederiksen T, Hesse J, Lehmann A, Yanai A. PESTO: Proactively secure distributed single sign-on, or how to trust a hacked server. In: Proc. of the 2020 IEEE European Symp. on Security and Privacy. Genoa: IEEE, 2020. 587–606. [doi: 10.1109/EuroSP48549.2020.00044]

[25] Wang D, Li WT, Wang P. Crytanalysis of three anonymous authentication schemes for multi-server environment. Ruan Jian Xue Bao/Journal of Software, 2018, 29(7): 1937–1952 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5361.htm [doi: 10.13328/j.cnki.jos.005361]

[26] Song YQ, Zhang Y, Xu CX, Li SY, Yang AJ, Cheng N. Edge-cloud-assisted certificate revocation checking: An efficient solution against irresponsible service providers. IEEE Internet of Things Journal, 2023, 10(17): 15563–15580. [doi: 10.1109/JIOT.2023.3264682]

[27] Song YQ, Xu CX, Zhang Y, Li SY. Hardening password-based credential databases. IEEE Trans. on Information Forensics and Security, 2024, 19: 469–484. [doi: 10.1109/TIFS.2023.3324326]

[28] Lamport L. Password authentication with insecure communication. Communications of the ACM, 1981, 24(11): 770–772. [doi: 10.1145/358790.358797]

[29] Chang CC, Wu TC. Remote password authentication with smart cards. IEE Proc. E (Computers and Digital Techniques), 1991, 138(3): 165–168. [doi: 10.1049/ip-e.1991.0022]

[30] Tsaur WJ. A flexible user authentication scheme for multi-server Internet services. In: Proc. of the 1st Int'l Conf. on Networking. Colmar: Springer, 2001. 174–183. [doi: 10.1007/3-540-47728-4_18]

[31] Alwen J, Chen B, Pietrzak K, Reyzin L, Tessaro S. Scrypt is maximally memory-hard. In: Proc. of the 36th Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Paris: Springer, 2017. 33–62. [doi: 10.1007/978-3-319-56617-7_2]

[32] Alwen J, Chen B, Kamath C, Kolmogorov V, Pietrzak K, Tessaro S. On the complexity of scrypt and proofs of space in the parallel random oracle model. In: Proc. of the 35th Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Vienna: Springer, 2016. 358–387. [doi: 10.1007/978-3-662-49896-5_13]

[33] Klein DV. "Foiling the cracker": A survey of, and improvements to, password security. In: Proc. of the 1990 USENIX Security Workshop. Boston, 1990. 5–14.

[34] Wei FS, Ma JF, Li GS, Ma CG. Efficient three-party password-based authenticated key exchange protocol in the standard model. Ruan Jian Xue Bao/Journal of Software, 2016, 27(9): 2389–2399 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4861.htm [doi: 10.13328/j.cnki.jos.004861]

[35] Everspaugh A, Chaterjee R, Scott S, Juels A, Ristenpart T. The pythia PRF service. In: Proc. of the 24th USENIX Conf. on Security Symp. Washington: USENIX Association, 2015. 547–562.

[36] Lai RWF, Egger C, Schröder D, Chow SSM. Phoenix: Rebirth of a cryptographic password-hardening service. In: Proc. of the 26th USENIX Conf. on Security Symp. Vancouver: USENIX Association, 2017. 899–916.

[37] Agrawal S, Miao PH, Mohassel P, Mukherjee P. PASTA: Password-based threshold authentication. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 2042–2059. [doi: 10.1145/3243734.3243839]

[38] Meadows C. A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In: Proc. of the 1986 IEEE Symp. on Security and Privacy. Oakland: IEEE, 1986. 134–137. [doi: 10.1109/SP.1986.10022]

[39] Freedman MJ, Nissim K, Pinkas B. Efficient private matching and set intersection. In: Proc. of the Advances in Cryptology—EUROCRYPT 2004. Interlaken: Springer, 2004. 1–19. [doi: 10.1007/978-3-540-24676-3_1]

[40] Chor B, Kushilevitz E, Goldreich O, Sudan M. Private information retrieval. Journal of the ACM, 1998, 45(6): 965–981. [doi: 10.1145/293347.293350]

[41] Goh EJ. Secure indexes. Cryptology ePrint Archive, 2003.

[42] Chang YC, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data. In: Proc. of the 3rd Int'l Conf. on Applied Cryptography and Network Security. New York: Springer, 2005. 442–455. [doi: 10.1007/11496137_30]

[43] Curtmola R, Garay J, Kamara S, Ostrovsky R. Searchable symmetric encryption: Improved definitions and efficient constructions. In: Proc. of the 13th ACM Conf. on Computer and Communications Security. Alexandria: ACM, 2006. 79–88. [doi: 10.1145/1180405.1180417]

[44] Li SY, Zhang Y, Xu CX, Cheng N, Liu Z, Du YC, Shen XM. HealthFort: A cloud-based ehealth system with conditional forward transparency and secure provenance via blockchain. IEEE Trans. on Mobile Computing, 2023, 22(11): 6508–6525. [doi: 10.1109/TMC.2022.3199048]

[45] He XY, Zhang Y, Li SY, Song YQ, Li HW. Privacy-driven fine-grained data trading. In: Proc. of the 34th Annual Int'l Symp. on

Personal, Indoor and Mobile Radio Communications. Toronto: IEEE, 2023. 1–6. [doi: 10.1109/PIMRC56721.2023.10294035]
[46] Yuan CH, Li Y, Ren S. Dynamic multi-keyword searchable encryption scheme. Chinese Journal of Network and Information Security, 2023, 9(2): 143–153 (in Chinese with English abstract). [doi: 10.11959/j.issn.2096-109x.2023028]
[47] Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G. Public key encryption with keyword search. In: Proc. of the Advances in Cryptology—EUROCRYPT 2004. Interlaken: Springer, 2004. 506–522. [doi: 10.1007/978-3-540-24676-3_30]
[48] Boneh D, Franklin M. Identity-based encryption from the Weil pairing. In: Proc. of the 21st Annual Int'l Cryptology Conf. on Advances in Cryptology—CRYPTO 2001. Santa Barbara: Springer, 2001. 213–229. [doi: 10.1007/3-540-44647-8_13]
[49] Chen YC. SPEKS: Secure server-designation public key encryption with keyword search against keyword guessing attacks. The Computer Journal, 2015, 58(4): 922–933. [doi: 10.1093/comjnl/bxu013]
[50] Li SY, Zhang Y, Song YQ, Cheng N, Yang K, Li HW. Blockchain-based portable authenticated data transmission for mobile edge computing: A universally composable secure solution. IEEE Trans. on Computers, 2024, 73(4): 1114–1125. [doi: 10.1109/TC.2024.3355759]
[51] Xu P, Jin H, Wu QH, Wang W. Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack. IEEE Trans. on Computers, 2013, 62(11): 2266–2277. [doi: 10.1109/TC.2012.215]
[52] Riva O, Qin C, Strauss K, Lymberopoulos D. Progressive authentication: Deciding when to authenticate on mobile phones. In: Proc. of the 21st USENIX Conf. on Security Symp. Bellevue: USENIX Association, 2012. 301–316.
[53] Chen RM, Mu Y, Yang GM, Guo FC, Huang XY, Wang XF, Wang YJ. Server-aided public key encryption with keyword search. IEEE Trans. on Information Forensics and Security, 2016, 11(12): 2833–2842. [doi: 10.1109/TIFS.2016.2599293]
[54] Bellare M, Keelveedhi S, Ristenpart T. DupLESS: Server-aided encryption for deduplicated storage. In: Proc. of the 22nd USENIX Conf. on Security. Washington: USENIX Association, 2013. 179–194.
[55] Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing. In: Proc. of the Advances in Cryptology—ASIACRYPT 2001. Gold Coast: Springer, 2001. 514–532. [doi: 10.1007/3-540-45682-1_30]
[56] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]
[57] Vo DL, Zhang FG, Kim K. A new threshold blind signature scheme from pairings. In: Proc. of the 2003 Symp. on Cryptography and Information Security Hamamatsu. 2003. 233–238.

附中文参考文献

[8] 俞东进, 王娇娇, 柳诚飞. 基于协作模式的工作流最优员工分配方法. 软件学报, 2018, 29(11): 3340–3354. http://www.jos.org.cn/1000-9825/5483.htm [doi: 10.13328/j.cnki.jos.005483]
[10] 王寅, 范铭, 陶俊杰, 雷靖薏, 晋武侠, 韩德强, 刘烃. 移动应用隐私权声明内容合规性检验方法. 软件学报, 2024, 35(8): 3668–3683. http://www.jos.org.cn/1000-9825/7121.htm [doi: 10.13328/j.cnki.jos.007121]
[21] 李经纬, 贾春福, 刘哲理, 李进, 李敏. 可搜索加密技术研究综述. 软件学报, 2015, 26(1): 109–128. http://www.jos.org.cn/1000-9825/4700.htm [doi: 10.13328/j.cnki.jos.004700]
[23] 张效林, 谷大武, 张驰. 移动平台典型应用的身份认证问题研究. 网络与信息安全学报, 2020, 6(6): 137–151. [doi: 10.11959/j.issn.2096-109x.2020081]
[25] 汪定, 李文婷, 王平. 对三个多服务器环境下匿名认证协议的分析. 软件学报, 2018, 29(7): 1937–1952. http://www.jos.org.cn/1000-9825/5361.htm [doi: 10.13328/j.cnki.jos.005361]
[34] 魏福山, 马建峰, 李光松, 马传贵. 标准模型下高效的三方口令认证密钥交换协议. 软件学报, 2016, 27(9): 2389–2399. http://www.jos.org.cn/1000-9825/4861.htm [doi: 10.13328/j.cnki.jos.004861]
[46] 袁承昊, 李勇, 任爽. 多关键词动态可搜索加密方案. 网络与信息安全学报, 2023, 9(2): 143–153. [doi: 10.11959/j.issn.2096-109x.2023028]

作者简介

张源, 博士, 研究员, CCF 专业会员, 主要研究领域为密码学, 信息安全, 区块链, 云计算安全.
陈曦露, 硕士生, 主要研究领域为隐私保护, 云计算安全.
宋雅晴, 硕士, 主要研究领域为数据安全, 应用密码学.
高歌, 硕士生, 主要研究领域为应用密码学, 隐私保护.
唐鹏, 博士, 助理研究员, 主要研究方向为人工智能安全, 隐私保护, 数据要素流通安全.
邱卫东, 博士, 教授, CCF 专业会员, 主要研究领域为计算机取证, 密码分析, 人工智能安全, 大数据隐私保护.
李洪伟, 博士, 教授, CCF 高级会员, 主要研究领域为网络空间安全, 密码技术.